ANLY 530 Final Project

Jia Zhao

8/1/2020

## Background

Competitiveness, market share, professional development and personal support to community action, health, culture, education and sport, are linked to a promising new market. Coupled with the development of organizations, the pressure to achieve goals more audacious, employees increasingly overwhelmed, they end up buying some disturbance in the health-related type of labor activity. The objective of this project is to apply some machine learning algorithms in the prediction of absenteeism at work. The database is the information collected records of absenteeism from work during the period of July/07 to July/2010 in a Courier company. Absences certified with the International Classification of Diseases were stratified into 21 categories, the data were tabulated and stored in two datasets (training and testing set). ## Object The objective of this project is to apply some machine learning algorithms in the prediction of absenteeism at work. Your job is to design a machine learning algorithm which tends to predict the absenteeism in hours. ## Task 1 - Mandatory Since the target variable is continuous, you should break it to some smaller sub groups: Group 0: Number of hours=0 Group 1: 0 < Number of hours <= 6 Group 2: Number of hours > 6

## Task 2- Optional (extra credit)

Predict the number of hours of absence without converting it to categorical variable (consider the continuous value)

## Loading libraries, Import Data, and Correlation

### Load Libraries

library(readr)  
library(caret) #train function for modeling, varImp

## Warning: package 'ggplot2' was built under R version 3.6.2

library(rattle)

## Warning: package 'rattle' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

library(party)

## Warning: package 'party' was built under R version 3.6.2

library(ggpubr)

## Warning: package 'ggpubr' was built under R version 3.6.2

library(ggplot2)  
library(GGally) #Scatterplot Matrix - ggpairs

## Warning: package 'GGally' was built under R version 3.6.2

library(dplyr)  
library(corrplot) # Scatterplot Matrix - corrplot  
library(ggcorrplot) # Scatterplot Matrix - ggcorrplot  
library(randomForest) # RandomForest model  
library(naivebayes)  
library(class) # KNN model  
library(rpart.plot)  
library(rpart)

## Import Data

data <-read\_csv("Absenteeism\_at\_work\_train.csv")

### Checking variables and data processing

str(data)

## tibble [666 × 21] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ ID : num [1:666] 11 36 3 7 11 3 10 20 14 1 ...  
## $ Reason for absence : num [1:666] 26 0 23 7 23 23 22 23 19 22 ...  
## $ Month of absence : num [1:666] 7 7 7 7 7 7 7 7 7 7 ...  
## $ Day of the week : num [1:666] 3 3 4 5 5 6 6 6 2 2 ...  
## $ Seasons : num [1:666] 1 1 1 1 1 1 1 1 1 1 ...  
## $ Transportation expense : num [1:666] 289 118 179 279 289 179 361 260 155 235 ...  
## $ Distance from Residence to Work: num [1:666] 36 13 51 5 36 51 52 50 12 11 ...  
## $ Service time : num [1:666] 13 18 18 14 13 18 3 11 14 14 ...  
## $ Age : chr [1:666] "33" "50" "38" "39" ...  
## $ Work load Average/day : num [1:666] 239554 239554 239554 239554 239554 ...  
## $ Hit target : num [1:666] 97 97 97 97 97 97 97 97 97 97 ...  
## $ Disciplinary failure : num [1:666] 0 1 0 0 0 0 0 0 0 0 ...  
## $ Education : num [1:666] 1 1 1 1 1 1 1 1 1 3 ...  
## $ Son : num [1:666] 2 1 0 2 2 0 1 4 2 1 ...  
## $ Social drinker : num [1:666] 1 1 1 1 1 1 1 1 1 0 ...  
## $ Social smoker : num [1:666] 0 0 0 1 0 0 0 0 0 0 ...  
## $ Pet : num [1:666] 1 0 0 0 1 0 4 0 0 1 ...  
## $ Weight : num [1:666] 90 98 89 68 90 89 80 65 95 88 ...  
## $ Height : num [1:666] 172 178 170 168 172 170 172 168 196 172 ...  
## $ Body mass index : num [1:666] 30 31 31 24 30 31 27 23 25 29 ...  
## $ Absenteeism time in hours : num [1:666] 4 0 2 4 2 2 8 4 40 8 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. ID = col\_double(),  
## .. `Reason for absence` = col\_double(),  
## .. `Month of absence` = col\_double(),  
## .. `Day of the week` = col\_double(),  
## .. Seasons = col\_double(),  
## .. `Transportation expense` = col\_double(),  
## .. `Distance from Residence to Work` = col\_double(),  
## .. `Service time` = col\_double(),  
## .. Age = col\_character(),  
## .. `Work load Average/day` = col\_number(),  
## .. `Hit target` = col\_double(),  
## .. `Disciplinary failure` = col\_double(),  
## .. Education = col\_double(),  
## .. Son = col\_double(),  
## .. `Social drinker` = col\_double(),  
## .. `Social smoker` = col\_double(),  
## .. Pet = col\_double(),  
## .. Weight = col\_double(),  
## .. Height = col\_double(),  
## .. `Body mass index` = col\_double(),  
## .. `Absenteeism time in hours` = col\_double()  
## .. )

data$Age <- as.numeric(data$Age)

## Warning: NAs introduced by coercion

### Checking for Missing or NA Values

df <- na.omit(data)  
df <- data.frame(df)

### Add a group column in the df

df <- df %>% mutate(  
 group = case\_when(  
 Absenteeism.time.in.hours == 0 ~ "0",  
 Absenteeism.time.in.hours >0 & Absenteeism.time.in.hours <= 6 ~ "1",  
 Absenteeism.time.in.hours >6 ~ "2"  
 )  
)  
df$group <- as.numeric(df$group)  
head(df)

## ID Reason.for.absence Month.of.absence Day.of.the.week Seasons  
## 1 11 26 7 3 1  
## 2 36 0 7 3 1  
## 3 3 23 7 4 1  
## 4 7 7 7 5 1  
## 5 11 23 7 5 1  
## 6 3 23 7 6 1  
## Transportation.expense Distance.from.Residence.to.Work Service.time Age  
## 1 289 36 13 33  
## 2 118 13 18 50  
## 3 179 51 18 38  
## 4 279 5 14 39  
## 5 289 36 13 33  
## 6 179 51 18 38  
## Work.load.Average.day Hit.target Disciplinary.failure Education Son  
## 1 239554 97 0 1 2  
## 2 239554 97 1 1 1  
## 3 239554 97 0 1 0  
## 4 239554 97 0 1 2  
## 5 239554 97 0 1 2  
## 6 239554 97 0 1 0  
## Social.drinker Social.smoker Pet Weight Height Body.mass.index  
## 1 1 0 1 90 172 30  
## 2 1 0 0 98 178 31  
## 3 1 0 0 89 170 31  
## 4 1 1 0 68 168 24  
## 5 1 0 1 90 172 30  
## 6 1 0 0 89 170 31  
## Absenteeism.time.in.hours group  
## 1 4 1  
## 2 0 0  
## 3 2 1  
## 4 4 1  
## 5 2 1  
## 6 2 1

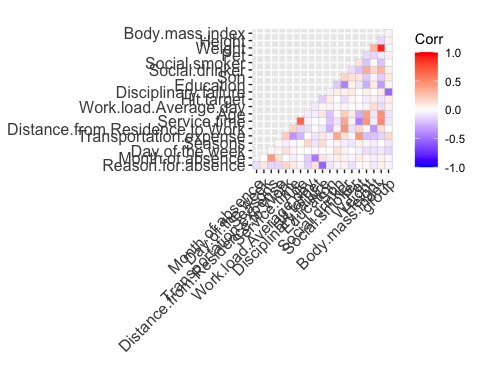
### Remove columns for the ID and Absenteeism.time.in.hours

df <- df[,-c(1,21)]  
head(df)

## Reason.for.absence Month.of.absence Day.of.the.week Seasons  
## 1 26 7 3 1  
## 2 0 7 3 1  
## 3 23 7 4 1  
## 4 7 7 5 1  
## 5 23 7 5 1  
## 6 23 7 6 1  
## Transportation.expense Distance.from.Residence.to.Work Service.time Age  
## 1 289 36 13 33  
## 2 118 13 18 50  
## 3 179 51 18 38  
## 4 279 5 14 39  
## 5 289 36 13 33  
## 6 179 51 18 38  
## Work.load.Average.day Hit.target Disciplinary.failure Education Son  
## 1 239554 97 0 1 2  
## 2 239554 97 1 1 1  
## 3 239554 97 0 1 0  
## 4 239554 97 0 1 2  
## 5 239554 97 0 1 2  
## 6 239554 97 0 1 0  
## Social.drinker Social.smoker Pet Weight Height Body.mass.index group  
## 1 1 0 1 90 172 30 1  
## 2 1 0 0 98 178 31 0  
## 3 1 0 0 89 170 31 1  
## 4 1 1 0 68 168 24 1  
## 5 1 0 1 90 172 30 1  
## 6 1 0 0 89 170 31 1

### Scatterplot Matrix

# sc <- ggpairs(df)  
# sc  
res <- round(cor(df),2)  
#ggcorrplot(res, type = "lower", ggtheme = ggplot2::theme\_gray, insig = "blank", lab =TRUE) #NAs also appear if there are attributes with zero variance (with all elements equal)  
ggcorrplot(res, type = "lower", ggtheme = ggplot2::theme\_gray, insig = "blank", lab =FALSE) #NAs also appear if there are attributes with zero variance (with all elements equal)



df <- df[,-11] # Rmove Disciplinary.failure

### Create Attribute Information Table

# Attribute\_information <- c("Certain infectious and parasitic diseases",  
# "Neoplasms",  
# "Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism",  
# "Endocrine, nutritional and metabolic diseases",  
# "Mental and behavioural disorders",  
# "Diseases of the nervous system",  
# "Diseases of the eye and adnexa",  
# "Diseases of the ear and mastoid process",  
# "Diseases of the circulatory system",  
# "Diseases of the respiratory system",  
# "Diseases of the digestive system",  
# "Diseases of the skin and subcutaneous tissue",  
# "Diseases of the musculoskeletal system and connective tissue",  
# "Diseases of the genitourinary system",  
# "Pregnancy, childbirth and the puerperium",  
# "Certain conditions originating in the perinatal period",  
# "Congenital malformations, deformations and chromosomal abnormalities",  
# "Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified",  
# "Injury, poisoning and certain other consequences of external causes",  
# "External causes of morbidity and mortality",  
# "Factors influencing health status and contact with health services",  
# "categories without (CID) patient follow-up",  
# "medical consultation",  
# "blood donation",  
# "laboratory examination",  
# "unjustified absence",  
# "physiotherapy",  
# "dental consultation"  
# )  
# Attribute\_information <- data.frame(Attribute\_information)  
# Attribute\_information <- tibble::rowid\_to\_column(Attribute\_information, "ID")  
# #Attribute\_information %>% mutate(id = row\_number())  
# Attribute\_information$Attribute\_information <- as.character(Attribute\_information$Attribute\_information)  
# head(Attribute\_information)

## Splitting the data

### splitting the data into training data (80%) and test data (20%)

set.seed(12345)  
df\_rand <- df[order(runif(662)),]  
df\_train <- df\_rand[1:529,]  
df\_test <- df\_rand[530:662,]  
  
prop.table(table(df\_rand$group))

##   
## 0 1 2   
## 0.05589124 0.58006042 0.36404834

prop.table(table(df\_train$group))

##   
## 0 1 2   
## 0.04347826 0.58601134 0.37051040

prop.table(table(df\_test$group))

##   
## 0 1 2   
## 0.1052632 0.5563910 0.3383459

Since group distribution of rand data, training data and test data look simiar, the randomization went well

## Logistic Regreesion Results

### Summary

fit\_log <- glm(group ~., data = df\_train)  
summary(fit\_log)

##   
## Call:  
## glm(formula = group ~ ., data = df\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.6071 -0.3126 -0.1749 0.5091 1.0430   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.444e+00 4.327e+00 1.258 0.208941   
## Reason.for.absence -5.595e-03 3.138e-03 -1.783 0.075192 .   
## Month.of.absence 1.250e-02 9.023e-03 1.385 0.166678   
## Day.of.the.week -6.037e-02 1.719e-02 -3.511 0.000486 \*\*\*  
## Seasons -5.194e-02 2.564e-02 -2.026 0.043293 \*   
## Transportation.expense 7.353e-04 4.998e-04 1.471 0.141830   
## Distance.from.Residence.to.Work -3.287e-04 2.595e-03 -0.127 0.899238   
## Service.time -3.364e-03 9.784e-03 -0.344 0.731081   
## Age -5.055e-03 5.886e-03 -0.859 0.390913   
## Work.load.Average.day 1.243e-06 5.987e-07 2.076 0.038376 \*   
## Hit.target 1.936e-02 7.223e-03 2.680 0.007605 \*\*   
## Education 4.468e-02 4.776e-02 0.936 0.349965   
## Son 3.150e-02 2.476e-02 1.272 0.203836   
## Social.drinker 6.617e-02 8.072e-02 0.820 0.412732   
## Social.smoker -1.509e-02 1.030e-01 -0.146 0.883591   
## Pet -1.234e-02 2.772e-02 -0.445 0.656443   
## Weight 4.120e-02 2.722e-02 1.514 0.130712   
## Height -3.618e-02 2.469e-02 -1.466 0.143360   
## Body.mass.index -1.109e-01 7.834e-02 -1.416 0.157342   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.2912097)  
##   
## Null deviance: 162.42 on 528 degrees of freedom  
## Residual deviance: 148.52 on 510 degrees of freedom  
## AIC: 869.25  
##   
## Number of Fisher Scoring iterations: 2

### Predict data

log\_prod <- predict(fit\_log, df\_test, type = "response")  
log\_pred <- rep(0, dim(df\_test)[1])  
log\_pred[log\_prod >.5] = 1  
log\_pred[log\_prod >1.5] = 2  
  
(p <- table(log\_pred, df\_test$group))

##   
## log\_pred 0 1 2  
## 1 9 68 30  
## 2 5 6 15

(Accuracy <- sum(diag(p))/sum(p)\*100)

## [1] 11.2782

## Random Forest

### Summary

random\_model <- randomForest(group ~., data = df\_train, ntree = 100, proximity = T)

## Warning in randomForest.default(m, y, ...): The response has five or fewer  
## unique values. Are you sure you want to do regression?

summary(random\_model)

## Length Class Mode   
## call 5 -none- call   
## type 1 -none- character  
## predicted 529 -none- numeric   
## mse 100 -none- numeric   
## rsq 100 -none- numeric   
## oob.times 529 -none- numeric   
## importance 18 -none- numeric   
## importanceSD 0 -none- NULL   
## localImportance 0 -none- NULL   
## proximity 279841 -none- numeric   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 11 -none- list   
## coefs 0 -none- NULL   
## y 529 -none- numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## terms 3 terms call

### Accuracy

random\_prod <- predict(random\_model, df\_test, type = "response")  
random\_pred <- rep(0, dim(df\_test)[1])  
random\_pred[random\_prod >.5] = 1  
random\_pred[random\_prod >1.5] = 2  
  
(p <- table(random\_pred, df\_test$group))

##   
## random\_pred 0 1 2  
## 0 10 0 0  
## 1 4 61 9  
## 2 0 13 36

(Accuracy <- sum(diag(p))/sum(p)\*100)

## [1] 80.45113

## Adding regression to trees

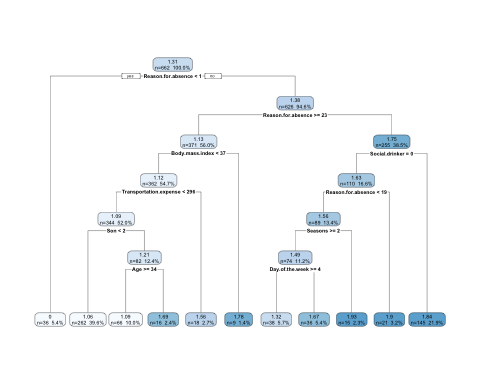
### Summary

df.rpart <- rpart(group ~. , data = df)  
df.rpart

## n= 662   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 662 215.1360000 1.308157   
## 2) Reason.for.absence< 0.5 36 0.0000000 0.000000 \*  
## 3) Reason.for.absence>=0.5 626 149.9872000 1.383387   
## 6) Reason.for.absence>=22.5 371 44.5283000 1.132075   
## 12) Body.mass.index< 37 362 39.1270700 1.116022   
## 24) Transportation.expense< 295.5 344 31.0232600 1.093023   
## 48) Son< 1.5 262 16.1412200 1.057252 \*  
## 49) Son>=1.5 82 13.4756100 1.207317   
## 98) Age>=33.5 66 5.4545450 1.090909 \*  
## 99) Age< 33.5 16 3.4375000 1.687500 \*  
## 25) Transportation.expense>=295.5 18 4.4444440 1.555556 \*  
## 13) Body.mass.index>=37 9 1.5555560 1.777778 \*  
## 7) Reason.for.absence< 22.5 255 47.9372500 1.749020   
## 14) Social.drinker< 0.5 110 25.7181800 1.627273   
## 28) Reason.for.absence< 18.5 89 21.9101100 1.561798   
## 56) Seasons>=1.5 74 18.4864900 1.486486   
## 112) Day.of.the.week>=3.5 38 8.2105260 1.315789 \*  
## 113) Day.of.the.week< 3.5 36 8.0000000 1.666667 \*  
## 57) Seasons< 1.5 15 0.9333333 1.933333 \*  
## 29) Reason.for.absence>=18.5 21 1.8095240 1.904762 \*  
## 15) Social.drinker>=0.5 145 19.3517200 1.841379 \*

### Graph

rpart.plot(df.rpart, digits = 3, fallen.leaves = TRUE, type =, extra =101)

 ## Navie Bayes

### Summary

df\_train$group <- as.character(df\_train$group)  
naive\_model <- naive\_bayes(group ~. , data = df\_train)   
summary(naive\_model)

##   
## ================================== Naive Bayes ==================================   
##   
## - Call: naive\_bayes.formula(formula = group ~ ., data = df\_train)   
## - Laplace: 0   
## - Classes: 3   
## - Samples: 529   
## - Features: 18   
## - Conditional distributions:   
## - Gaussian: 18  
## - Prior probabilities:   
## - 0: 0.0435  
## - 1: 0.586  
## - 2: 0.3705  
##   
## ---------------------------------------------------------------------------------

### Accuracy

naive\_prod <- predict(naive\_model, newdata = df\_test)  
  
(p <- table(naive\_prod, df\_test$group))

##   
## naive\_prod 0 1 2  
## 0 13 0 0  
## 1 1 57 12  
## 2 0 17 33

(Accuracy <- sum(diag(p))/sum(p)\*100)

## [1] 77.44361

## KNN

### Summary

df\_train$group <- as.factor(df\_train$group)  
df\_test$group <- as.factor(df\_test$group)  
df\_train\_labels <- df\_train[1:529,19]  
KNN\_model <- knn(train = df\_train, test = df\_test, cl=df\_train\_labels, k = 13)  
summary(KNN\_model)

## 0 1 2   
## 1 104 28

### Accuracy

(p <- table(KNN\_model, df\_test$group))

##   
## KNN\_model 0 1 2  
## 0 0 1 0  
## 1 10 63 31  
## 2 4 10 14

(Accuracy <- sum(diag(p))/sum(p)\*100)

## [1] 57.89474